

# Enumération des partitions de diamètre minimum

A. Guénoche

*G.R.T.C.–C.N.R.S., 31 Ch. J. Aiguier, 13402 Marseille Cedex 9, France*

Received 22 July 1991

## Résumé

Guénoche, A., Enumération des partitions de diamètre minimum, *Discrete Mathematics* 111 (1993) 277–287.

Nous énumérons les  $p$ -partitions des sommets d'un graphe valué (partitions à nombre de classes  $p$  fixé) de diamètre  $\Delta$  minimum. Dans le cas où  $p=2$ , nous rappelons la construction de M.R. Rao qui permet de les dénombrer et de les énumérer. Pour  $p>2$  le problème revient à construire un graphe seuil minimum  $G_\sigma$  dont les arêtes ont une longueur supérieure à  $\Delta$ , et qui soit  $p$ -coloriable. C'est donc un problème NP-difficile. Nous utilisons des heuristiques pour déterminer une borne supérieure  $\sigma$  de  $\Delta$  et nous énumérons les  $p$ -colorations de  $G_\sigma$ . Ensuite on fait décroître  $\sigma$  tant qu'il existe au moins une coloration compatible. Quand l'algorithme s'arrête, le diamètre des partitions restantes est la plus grande valeur inférieure au seuil obtenu.

## Abstract

Guénoche, A., Enumération des partitions de diamètre minimum, *Discrete Mathematics* 111 (1993) 277–287.

We enumerate all the  $p$ -partitions of vertices of a complete valued graph (partitions with  $p$  classes with minimum diameter  $\Delta$ ). When  $p=2$ , we recall M.R. Rao's algorithm that permits to count and to enumerate bipartitions. When  $p>2$ , the problem is the same as to build a minimum threshold spanning subgraph  $G_\sigma$  having edges greater than  $\Delta$  and being  $p$ -colourable. So it is an NP-hard problem. First, we use some heuristics to determine  $\sigma$ , an upper approximation of  $\Delta$ , to enumerate all the  $p$  colourings of  $G_\sigma$ . Next we reduce  $\sigma$  until the partitions remain compatible. When algorithm stops, the diameter of the remaining partitions is the greatest length of an edge lower than the threshold  $\sigma$ .

## 0. Introduction

Soit  $X = \{x_1, x_2, \dots, x_n\}$  un ensemble de  $n$  objets et  $D = \{d_{ij}\}$  un tableau de dissimilarités sur  $X$ .

$$d_{ij} = d(x_i, x_j) = d(x_j, x_i) \geq 0, \quad d_{ii} = 0.$$

*Correspondence to:* A. Guénoche, G.R.T.C., C.N.R.S., 31 Ch. J. Aiguier, 13402 Marseille Cedex 9, France.

Cette donnée est équivalente au graphe complet  $G=(X, X \times X, D)$  de sommets  $X$  et dont les arêtes sont valuées par  $D$ ;  $d_{ij}$  est également appelée la longueur de l'arête  $(x_i, x_j)$ . Dans la suite on appelle graphe (au) seuil  $\sigma$  le graphe partiel  $G_\sigma=(X, U)$  dont les seules arêtes sont celles de longueur  $\geq \sigma$ . On note  $U_\sigma$  l'ensemble des arêtes de  $G$  dont la longueur est égale à  $\sigma$ .

Soit  $P=\{C_1, C_2, \dots, C_p\}$  une  $p$ -partition, c'est à dire une partition de  $X$  en  $p$  classes.

$$C_i \cap C_j = \emptyset, \quad \bigcup_{i=1, \dots, p} C_i = X.$$

Rappelons que dans un graphe valué:

- le diamètre d'une classe est égal à la plus grande dissimilarité intra-classe,

$$\delta(C_k) = \max d_{ij} \quad \text{pour } x_i \in C_k \quad \text{et } x_j \in C_k.$$

- le diamètre d'une partition est égal au plus grand des diamètres de ses classes,

$$\Delta(P) = \max_{k=1, \dots, p} \delta(C_k).$$

On s'intéresse aux partitions de diamètre minimum de préférence à celles de somme des diamètres des classes minimum, car ce dernier critère conduit le plus souvent à choisir  $p-1$  classes réduites à un singleton, donc de diamètre nul, et de ce fait est d'un intérêt pratique limité. Construire sur  $X$  une partition  $P$  telle que  $\Delta(P)$  soit minimum est un problème dont la complexité dépend du nombre de classes.

Pour  $p=2$ , Rao [10] a donné un algorithme de complexité  $O(n^2 \log n)$  qui permet de dénombrer et d'énumérer toutes les partitions en deux classes de diamètre minimum. Dans la lignée des travaux de Hubert [7], Leclerc [8] et Hansen et Jaumard [6] sur les méthodes de classification descendantes, Monma et Suri [9] d'une part et Guénoche [3] d'autre part, on a proposé un autre algorithme de complexité  $O(n^2)$ , basé sur la bicoloration d'un arbre maximum pour construire une (seule) bipartition de diamètre minimum. Soulignons de plus que toutes les partitions de diamètre minimum ne sont pas des bi-colorations d'un arbre maximum.

Pour  $p>2$ , Hansen et Delattre [5] ont montré que le problème est NP-difficile. Ils proposent un algorithme basé sur une méthode de coloration optimale de graphes seuils successifs. En adaptant l'algorithme de Brown [2], ils construisent pour chaque valeur de  $p$  une seule partition de diamètre minimum.

Notre propos est d'énumérer, pour  $p$  fixé, les  $p$ -partitions de diamètre minimum. Dans le cas où  $p=2$ , nous rappelons la construction de Rao. Pour  $p>2$  nous présentons un algorithme qui permet de les construire toutes. Comme pour la méthode de Hansen et Delattre cet algorithme est basé sur la coloration d'un graphe seuil avec  $p$  couleurs, chaque couleur définissant une classe. On utilise plusieurs heuristiques pour approximer le diamètre et on n'énumère les partitions qu'en dernier recours.

## 1. La méthode de Rao

Rappelons brièvement l'algorithme de Rao: On considère les arêtes dans l'ordre des longueurs décroissantes. Soit  $(x_i, x_j)$  une arête telle que  $d_{ij}$  est maximum. Tant que l'on peut séparer  $x_i$  et  $x_j$ , on réalise une partition de diamètre inférieur à  $d_{ij}$ . De trois choses l'une:

Si seul l'un des deux sommets est classé, on placera l'autre dans l'autre classe.

Si les deux le sont, dans des classes différentes, on passera à l'arête suivante. S'ils sont dans une même classe, l'algorithme s'arrête et  $d_{ij}$  est le diamètre de la (des) partition(s) obtenue(s).

Si aucun n'est classé, il suffit de noter le fait qu'ils doivent être dans des classes différentes. Pour n'examiner les arêtes qu'une seule fois, l'astuce de Rao consiste à donner à chaque sommet une étiquette. Ces labels peuvent être soit définitifs,  $n+1$  ou  $n+2$  correspondant aux 2 classes, soit temporaires dans l'intervalle entier  $[-n, n]$ ; deux sommets qui doivent être séparés reçoivent des labels opposés. L'examen de chaque arête conduit à une mise à jour des labels. Il sont rangés dans un tableau  $L$  indexé sur  $X$  initialisé à 0.

### Algorithme de Rao

Soit  $(u, v)$  une arête de valeur maximum; on pose  $L(u) := n+1$  et  $L(v) := n+2$

Répéter

Soit  $(i, j)$  une nouvelle arête de valeur maximum.

Si  $L(i) = L(j) = 0$  Alors  $L(i) := i$ ;  $L(j) := -i$

Sinon si  $L(i) = n+1$  (resp.  $n+2$ ) Et  $L(j) = 0$  Alors  $L(j) := n+2$  (resp.  $n+1$ )

Sinon si  $L(i) = k \leq n$  Et  $L(j) = 0$  Alors  $L(j) := -k$

Sinon si  $L(i) = n+1$  (resp.  $n+2$ ) Et  $L(j) = k$  Faire

Pour tout  $v$  tel que  $L(v) = k$  Faire  $L(v) := n+2$  (resp.  $n+1$ )

Pour tout  $v$  tel que  $L(v) = -k$  Faire  $L(v) := n+1$  (resp.  $n+2$ )

Sinon si  $L(i) = k \leq n$  Et  $L(j) = m$  Avec  $k \neq m \leq n$  Faire

Pour tout  $v$  tel que  $L(v) = m$  Faire  $L(v) := -k$

Pour tout  $v$  tel que  $L(v) = -m$  Faire  $L(v) := k$

Sinon si  $L(i) = L(j)$  Fin.

Une structure de données plus élaborée, décrite dans Guénoche, Hansen et Jaumard [4] permet de montrer que cet algorithme est de complexité  $O(n^2 \log n)$  dans le cas le pire. De toute manière il faut préalablement ordonner les arêtes dans l'ordre décroissant, ce qui donne une complexité du même ordre.

Si après exécution de cet algorithme il reste  $k$  étiquettes positives ou nulles, il y a  $2^k$  bipartitions de diamètre minimum. En effet, chacune de ces étiquettes peut recevoir n'importe quelle valeur,  $n+1$  ou  $n+2$ ; les étiquettes négatives prendront l'autre valeur. Dans le cas le pire, si  $D$  est injective, les trois plus longues arêtes forment un

cycle; la plus petite définit le diamètre des  $2^{n-3}$  partitions de diamètre minimum. Si  $D$  n'est pas injective, il y en a  $2^{n-2}$ .

## 2. Partitions en $p$ classes de diamètre minimum

Avant de présenter les grandes lignes de notre méthode, rappelons qu'un graphe est  $p$ -coloriable si l'on peut donner à chaque sommet une couleur parmi  $p$ , de façon que toute arête lie deux sommets de couleurs différentes. La plus petite valeur  $p$  pour laquelle un graphe est  $p$ -coloriable est appelé le *nombre chromatique* du graphe (donc le graphe n'est pas  $(p-1)$ -coloriable). Comme l'ont remarqué Hansen et Delattre [5], il existe une  $p$ -partition de diamètre inférieur à  $\sigma$  si et seulement si  $G_\sigma$  est  $p$ -coloriable.

Pour construire une  $p$ -partition de diamètre minimum, il suffit de trouver un plus petit seuil  $\sigma$  tel que  $G_\sigma$  soit  $p$ -coloriable. C'est ce qu'ont vu et appliqué Hansen et Delattre en considérant tous les graphes seuils de  $G$  quand  $\sigma$  décroît. Décrivons rapidement leur méthode: Initialement tous les sommets sont dans une même classe et on considère les arêtes dans l'ordre décroissant. On ajoute à chaque itération dans  $G_\sigma$  toutes les arêtes de longueur égale. Soit chacune de ces arêtes lie deux sommets de classes différentes; alors on passe au seuil suivant, soit l'une d'elle lie deux sommets d'une même classe. Dans ce cas on applique un algorithme de coloration optimal. Si l'on doit ajouter une  $(p+1)$ ème couleur pour colorier  $G_\sigma$ , la valeur précédente du seuil et la  $p$ -coloration trouvée est une  $p$ -partition de diamètre minimum. Leur méthode est donc un algorithme de coloration optimale itéré sur les graphes seuils obtenus en ajoutant des arêtes. Pour chaque valeur de  $p$ , il ne donne qu'une seule  $p$ -partition, or:

**Proposition 2.1.** *Le nombre de  $p$ -partitions de diamètre minimum d'un graphe à  $n$  sommets est au plus  $p^{n-p}$ .*

**Démonstration.** Considérons un graphe complet valué à  $n$  sommets dont les  $p(p-1)/2$  plus grandes arêtes forment une clique à  $p$  éléments. Si les  $p$  arêtes suivantes (dans l'ordre décroissant) de valeurs égales  $\Delta$ , lient un même sommet hors clique à chaque sommet de la clique, ce sommet et tous les autres peuvent être placés dans n'importe laquelle des  $p$  classes qui seront toutes de diamètre  $\Delta$ . Il a  $p^{n-p}$  possibilités.  $\square$

Néanmoins, sur des données réelles dans lesquelles les distances mesurent des proximités entre objets, et si sur ceux-ci existe des structures objectives de classes, on observe un nombre raisonnable de partitions; du moins on peut essayer de les énumérer si leur nombre ne dépasse pas une quantité fixée, au delà de laquelle cette énumération n'a aucun sens.

### 3. Énumération des $p$ -partitions ( $p > 2$ ) de diamètre minimum

Pour énumérer les  $p$ -partitions de diamètre minimum à plus de deux classes, on applique un algorithme en trois étapes:

- On utilise une méthode heuristique pour déterminer un seuil  $\sigma$ , approximation supérieure de  $\Delta$ , tel que  $G_\sigma$  soit  $p$ -coloriable. On détermine  $\sigma$  par une suite de subdivisions dichotomiques de l'intervalle de variation des dissimilarités. Pour chaque essai on utilise un algorithme glouton de coloration en  $O(n^2)$  du à Brélaz [1] basé sur la saturation. Nous l'améliorons légèrement par un choix différent de la couleur à donner au sommet le plus saturé. Cet algorithme construit un ordre de coloration des sommets.

- Ce seuil  $\sigma$  étant fixé, on énumère toutes les colorations de  $G_\sigma$  en  $p$  couleurs. On utilise une clique de  $G_\sigma$ , pas nécessairement maximum, obtenue par l'algorithme précédent, dont chaque sommet est le représentant d'une classe. Puis on colore les autres sommets, dans l'ordre de coloration. Si un sommet est adjacent à des sommets coloriés, ces couleurs lui sont interdites et toutes les autres sont possibles. En développant une procédure branch and bound, on obtient ainsi toutes les partitions en  $p$  classes dont le diamètre est inférieur à  $\sigma$ .

- Enfin on considère les arêtes dans l'ordre décroissant des valeurs de dissimilarités, à partir de  $\sigma$ . Une arête peut être ajoutée tant qu'il existe une partition compatible, dans laquelle cette arête lie deux sommets de classes différentes. Chaque arête élimine ainsi certaines des partitions précédemment obtenues. La première arête qui ne peut être retenue, parce qu'alors il ne resterait plus de partitions compatibles, a pour valeur  $\Delta$  qui est la plus grande des dissimilarités intra-classe. Les partitions restantes sont de diamètre minimum (pour un nombre de classes fixé). Cette étape est d'autant plus rapide que  $\sigma$  était une bonne approximation de la valeur de  $\Delta$  cherchée.

#### 3.1. Graphe seuil $p$ -coloriable

En présentant notre algorithme dans ses grandes lignes, nous avons annoncé une première étape qui construit un graphe  $p$ -coloriable dont les arêtes sont celles qui ont une valeur supérieure ou égale à un seuil à déterminer. Pour ce faire nous utilisons une méthode de subdivision dichotomique et une méthode séquentielle de coloration.

#### Méthode de subdivision

$D_{\min} := \min D_{ij}$ ;  $D_{\max} := \max D_{ij}$

Tant que  $D_{\max} - D_{\min} > 2$  Faire

$\sigma := (D_{\min} + D_{\max})/2$

On construit  $G_\sigma$  et on le colore par  $D_{\text{satur}}$

Soit  $NbCoul$  le nombre de couleurs obtenues

Si  $NbCoul \leq p$  Alors  $D_{\max} := \sigma$  Sinon  $D_{\min} := \sigma$

Les algorithmes séquentiels de coloration colorient les sommets dans un certain ordre. Dans l'algorithme Dsatur de Brélaz [1], on part d'un sommet de degré maximum auquel on donne la couleur 1. Tous les sommets qui lui sont adjacents sont *saturés* pour cette couleur. On définit un indice de saturation, pour chaque sommet non colorié, égale au nombre de couleurs différentes auquel il est adjacent. A chaque itération on choisit de colorier le sommet le plus saturé et en cas d'égalité, on retient celui qui est de degré maximum dans le sous-graphe partiel des sommets non coloriés. On lui donne la plus petite couleur possible. Brélaz a montré que pour les graphes bipartis, cet algorithme est optimal.

Nous proposons une modification de Dsatur. A chaque étape on colorie le même sommet, ce qui fait que l'on ne change rien à l'optimalité de la procédure pour les graphes bipartis. Mais à la suite de nombreuses expérimentations nous avons observé qu'il était préférable de donner, à ce sommet le plus saturé, la couleur qui sature le plus petit nombre de sommets non coloriés, insaturés pour cette couleur. Ceci revient à laisser libre plus de couleurs pour les sommets qui seront coloriés ultérieurement et la complexité de la procédure reste la même.

**Exemple 3.1.** On commence avec un sommet de degré maximum,  $x_3$  et ses 5 sommets adjacents sont saturés pour la couleur 1. Parmi ceux-ci  $x_2$  est de degré maximum; il reçoit la couleur 2 et ses 4 sommets adjacents sont saturés pour la couleur 2. Parmi les sommets saturés pour les deux couleurs, on choisit  $x_6$  qui est de degré maximum et on lui donne la couleur 3. Il n'y a plus, pour l'instant, de nouvelle couleur qu'il est nécessaire d'ajouter, puisqu'aucun sommet n'est adjacent aux trois sommets coloriés. On continue de colorier les sommets les plus saturés et de degré maximum dans le sous-graphe partiel des sommets non coloriés, donc  $x_9$  est de couleur 3 ainsi que  $x_5$ , ce qui sature  $x_1$  pour cette couleur. Jusqu'à présent il n'y a eu aucun choix; On obtient la table de saturation ci-dessous dans laquelle une  $x$  désigne la couleur du sommet et un 1 indique les sommets saturés par cette couleur.

Sommets	1	2	3	4	5	6	7	8	9
Degrés	3	4	5	3	2	4	4	5	4
Couleur 1		1	$x$		1	1	1		1
Couleur 2		$x$	1		1	1			1
Couleur 3	1	1	1	1	$x$	$x$		1	$x$

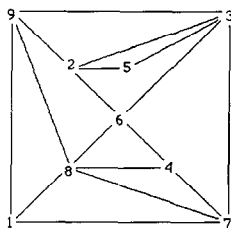


Fig. 1. Un graphe 3-coloriable.

Dans le graphe partiel restant, les sommets  $x_1$  et  $x_4$  sont de degré 2 et  $x_7$  et  $x_8$  sont de degré 3. Il faut donc colorier  $x_7$  et on a le choix entre les couleurs 2 et 3.

– L'algorithme classique lui donnerait la première ce qui saturerait  $x_1, x_4$  et  $x_8$  pour la couleur 2. En conséquence,  $x_8$  ne peut recevoir que la couleur 1, ce qui sature  $x_1$  et  $x_4$  à qui il ne reste aucune couleur. On aboutit donc suivant cette procédure à une coloration en 4 couleurs:

$$\{x_3, x_8 \mid x_2, x_7 \mid x_5, x_6, x_9 \mid x_1, x_4\}.$$

– Maintenant, suivant notre modification, on préférera pour colorier  $x_7$  la couleur 3, car  $x_1, x_4$  et  $x_8$  sont déjà saturés pour cette couleur. Pour  $x_8$  qui sature  $x_1$  et  $x_4$  le choix est indifférent puisqu'aucun de ces sommets n'est saturé pour les couleurs 1 ou 2 et si l'on donne à  $x_8$  la couleur 1,  $x_1$  et  $x_4$  recevront la couleur 2. On aboutit à une coloration en 3 couleurs:

$$\{x_3, x_8 \mid x_1, x_2, x_4 \mid x_5, x_6, x_7, x_9\}.$$

Nous avons montré sur cet exemple que cette modification, somme toute naturelle, pouvait être efficace; mais nous aurions pu également montrer sur un autre exemple qu'elle pouvait être pernicieuse. Pour trancher nous avons fait de nombreuses expérimentations en tirant au hasard des graphes à nombre de sommets  $N$  et nombre d'arêtes fixés. Les nombres d'arêtes choisis correspondent à une densité  $D$  d'arêtes égale à la proportion d'arêtes du graphe complet à  $N$  sommets. Nous en reproduisons les résultats ci-dessous. Pour chaque valeur de  $N$  et de  $D$  les résultats portent sur 100 graphes (ce sont des moyennes calculées sur 300 graphes). Le premier chiffre indique le nombre de graphes pour lesquels notre modification s'est appliquée, c'est à dire qu'il y avait plusieurs couleurs possibles pour un sommet et ce n'est pas la première qui a été retenue. Le second indique le nombre de graphes pour lesquels l'algorithme modifié a conduit à un plus petit nombre de couleurs, le troisième ceux pour lesquels l'algorithme classique donne un plus petit nombre. Dans les autres cas les deux méthodes ont donné des résultats identiques.

$N = 20$	$D = 30$	56	5	1
	$D = 50$	59	6	0
	$D = 70$	43	2	0
$N = 30$	$D = 30$	73	9	2
	$D = 50$	67	10	3
	$D = 70$	55	7	3
$N = 40$	$D = 30$	78	10	4
	$D = 50$	80	8	5
	$D = 70$	70	11	2
$N = 50$	$D = 30$	89	16	7
	$D = 50$	84	14	4
	$D = 70$	70	8	5

On remarque d'abord que la modification s'applique d'autant plus souvent que le graphe a de sommets et ensuite que le nombre de graphes pour lesquels elle s'avère bénéfique est toujours supérieur au nombre de cas où elle s'avère néfaste. Nous l'adopterons donc par la suite.

Il est clair qu'au début de cet algorithme on construit une clique, pas nécessairement maximale. En effet le premier sommet étant coloré, on colorie avec une nouvelle couleur un sommet qui lui est adjacent, puis si il y a un sommet adjacent à ces deux là il reçoit une troisième couleur, et ainsi de suite. Si le cardinal NbQ de cette clique est le nombre de couleurs obtenu en fin d'algorithme, cette coloration est optimale. C'est le cas dans notre exemple puisque  $\{x_3, x_2, x_6\}$  est une clique. A la suite on a considéré les sommets dans un certain ordre, ici  $(x_3, x_2, x_6, x_9, x_5, x_7, x_8, x_4, x_1)$  qui est l'ordre dans lequel les sommets sont coloriés dans la procédure d'énumération ci-dessous.

### 3.2. Enumération des $p$ -colorations de $G_\sigma$

On dispose maintenant d'un graphe seuil  $p$ -coloriable, d'un ordre sur les sommets dont les NbQ premiers forment une clique. Pour construire toutes ses  $p$ -colorations, on utilise cette clique comme initialisation du processus d'énumération. Nous allons développer une structure arborescente, en largeur d'abord, dans laquelle chaque nœud désigne un sommet du graphe et une couleur associée à ce sommet. Dans cette arborescence un chemin entre une feuille et la racine est un début de coloration dans laquelle un sommet ne peut apparaître qu'une seule fois. On initialise cette arborescence par une chaîne constituée des NbQ sommets de la clique  $Cl(.)$  ayant les couleurs  $1, 2, \dots, NbQ$ . A chaque itération on prolonge toutes les feuilles de plus grande profondeur. Elles sont toutes associées à un même sommet  $s$ . On colorie le sommet  $v$  qui suit  $s$  dans l'ordre de coloration. Pour chaque feuille, en remontant dans l'arborescence on marque les couleurs des sommets (coloriés) adjacents à  $x_i$ ; elles lui sont interdites. Toutes les autres couleurs donnent lieu à une nouvelle feuille de l'arborescence. Quand tous les sommets sont coloriés, toutes les feuilles et les chemins associés de l'arborescence correspondent à des colorations différentes.

Nous avons choisi une structure de données très simple, constituée de trois tableaux  $V, C, P$  indexés sur les sommets de l'arborescence:  $V(i)$  contient le numéro d'un sommet de  $G_\sigma$ ,  $C(i)$  sa couleur. L'indice  $i$  désigne le  $i$ -ième sommet de l'arborescence et  $P(i)$  est le numéro du prédécesseur de  $i$  dans l'arborescence. Le sommet suivant  $s$  dans l'ordre de Dsatur est codé  $Suc(s)$ . Le dernier sommet à colorier est appelé  $Der$ .

#### Enumération des $p$ -partitions de diamètre $< \sigma$

Pour  $i := 1$  à NbQ Faire

$$V(i) := Cl(i); \quad C(i) := i; \quad P(i) := i - 1$$



Tous les sommets de la clique sont classés

```

    i1 := NbQ;    i2 := NbQ
Tant que i1 ≤ i2 Faire
    s := V(i1);    v := Suc(s)    On essaye de colorier s
    Pour c := 1 à p Faire
        Coul(c) := 0    Les p couleurs sont possibles
    k := i1
    Tant que k ≠ 0 Faire
        t := V(k); Si t est adjacent à v Faire Coul(C(k)) := 1
        k := P(k)
    Pour c := 1 à p Faire
        Si Coul(c) = 0 Faire    La couleur c reste possible
            Si v ≠ Der Faire i2 := i2 + 1; V(i2) := v; C(i2) := c; P(i2) := i1
            Sinon Faire Coul(v) := c; Sortir la partition
    
```

Cet algorithme a une complexité proportionnelle au nombre de  $p$ -colorations du graphe  $G_\sigma$ . Plus le seuil  $\sigma$  est petit, plus le graphe possède d'arêtes, et moins il a de colorations. C'est pourquoi nous avons prolongé la recherche dichotomique sans nous contenter du premier seuil qui permette une  $p$ -coloration.

**Remarque.** Cet algorithme fonctionne pour toute valeur de  $p$ . Ici en donnant à  $p$  le nombre de classes déterminé par une méthode gloutonne, on est assuré de trouver au moins une partition. Si l'on avait choisi une valeur plus petite  $k < p$ , de deux choses l'une: Soit on trouve une partition en  $k$  classes de  $G_\sigma$ , soit on aboutit à un échec, c'est à dire qu'il n'y a plus de couleurs possibles pour un sommet (elles sont toutes marquées).

### 3.3. Elimination des partitions qui ne sont pas de diamètre minimum

On a jusqu'à présent construit toutes les partitions de diamètre inférieur au seuil  $\sigma$  qui définit notre graphe  $G_\sigma$ . On va à nouveau considérer les arêtes de  $U_{\sigma'}$  dont la longueur est immédiatement inférieure à  $\sigma$ . Si il existe au moins une partition telle que pour toute arête  $(x_i, x_j)$ , les sommets  $x_i$  et  $x_j$  ne sont pas dans la même classe, il existe donc une partition de diamètre  $d_{ij} < \sigma'$ . Il faut donc que toutes les arêtes de  $U_{\sigma'}$  satisfassent à cette condition. Les partitions restantes sont de diamètre inférieur à  $\sigma'$ . Cette procédure constitue la troisième étape de notre algorithme.

On considère toutes les arêtes de valeur  $\Delta$  immédiatement inférieure à  $\sigma$ . Si il existe au moins une partition compatible avec ces arêtes, toutes les partitions compatibles sont de diamètre inférieure à  $\sigma$ ; on itère la procédure en posant  $\sigma := \Delta$  et en ne gardant que les partitions compatibles. Si aucune partition n'est comptible avec toutes les arêtes de valeur  $\Delta$ , l'algorithme s'arrête.

**Exemple 3.2.** Considérons le tableau de dissimilarité ci-dessous, et construisons les partitions en trois classes de diamètre minimum:

	1	2	3	4	5	6	7	8
2	12							
3	65	10						
4	86	80	93					
5	72	70	53	66				
6	79	53	56	45	54			
7	7	97	67	33	42	51		
8	49	32	70	15	5	56	66	
9	82	80	74	73	95	74	23	46

Les colorations des graphes seuil par une succession de choix dichotomiques donnent 5 couleurs au seuil 51, 3 couleurs au seuil 74, 4 couleurs au seuil 62, 3 au seuil 68, 4 au seuil 65 et finalement 3 couleurs au seuil 67 (Fig. 2). La partition obtenue à ce seuil est  $\{x_7, x_8, x_9 | x_1, x_2, x_3 | x_4, x_5, x_6\}$  et l'ordre d'examen des sommets est  $(x_9, x_1, x_4, x_2, x_3, x_5, x_7, x_6, x_8)$ ; les trois premiers forment une clique.

On énumère les colorations de ce graphe par 3 couleurs. Les sommets  $x_2$  et  $x_3$  adjacents à  $x_4$  et  $x_9$  sont nécessairement dans la classe de  $x_1$ . Le sommet  $x_5$  adjacent à trois sommets classés n'a qu'une seule classe possible, celle de  $x_4$ . Le sommet  $x_7$  a deux possibilités la classe de  $x_4$  ou celle de  $x_9$ ,  $x_6$  est nécessairement dans celle de  $x_4$ . Pour finir,  $x_8$  peut être indifféremment dans la classe de  $x_9$  ou celle de  $x_4$ . On a donc 4 partitions de diamètre strictement inférieur à 67:

$$\{x_7, x_8, x_9 | x_1, x_2, x_3 | x_4, x_5, x_6\},$$

$$\{x_8, x_9 | x_1, x_2, x_3 | x_4, x_5, x_6, x_7\},$$

$$\{x_7, x_9 | x_1, x_2, x_3 | x_4, x_5, x_6, x_8\}$$

et

$$\{x_9 | x_1, x_2, x_3 | x_4, x_5, x_6, x_7, x_8\}.$$

Dans la troisième étape on examine les arêtes de longueur immédiatement inférieure;  $(x_7, x_8)$  qui est compatible avec la seconde et la troisième partitions ci-dessus, et  $(x_4, x_5)$  qui n'est compatible avec aucune. Les quatre partitions sont donc de diamètre 66 et ce sont les seules.

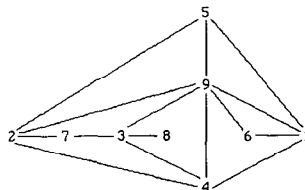


Fig. 2. Le graphe  $G_{67}$ .

#### 4. Conclusion

Indépendamment des partitions à nombre de classes fixé et de diamètre minimum, on peut adapter cet algorithme au calcul du nombre chromatique d'un graphe quelconque. On commence par appliquer l'algorithme *Dsatur* qui donne une coloration gloutonne en  $NbCoul$  couleurs et détermine une clique à  $NbQ$  sommets. Si ces deux valeurs sont égales, le nombre chromatique est atteint. Sinon on peut essayer une énumération des colorations à  $k$  couleurs, identique à celle des  $k$ -partitions du graphe seuil, pour toutes les valeurs de  $k$  variant de  $NbQ$  jusqu'à  $NbCoul - 1$ . La plus petite valeur de  $k$  qui aboutit est l'indice chromatique, et si l'on aboutit pas à une coloration en  $NbCoul - 1$  couleurs, c'est que  $NbCoul$  est le nombre chromatique.

Cet algorithme est rapide, mais 'gourmand' en place mémoire, puisque l'on développe une arborescence en largeur d'abord; même en acceptant jusqu'à 30 000 nœuds dans l'arborescence, il se peut qu'elle soit insuffisante pour un graphe à 25 sommets. Au delà il est nécessaire d'appliquer une procédure de 'backtracking' à la manière de Brown [2].

#### References

- [1] D. Brélaz, New methods to color the vertices of a graph, *Comm. ACM* 22(4) (1979) 251–256.
- [2] J.R. Brown, Chromatic scheduling and the chromatic number problem, *Management Sci.* 19 (1972) 456–463.
- [3] A. Guénoche, Partitions with minimum diameter, *Conf. IFCS*, Charlottesville, 1989.
- [4] A. Guénoche, P. Hansen and B. Jaumard, Efficient algorithms for divisive hierarchical clustering with the diameter criterion, *J. Classification* 8 (1991) 5–30.
- [5] P. Hansen and M. Delattre, Complete-link cluster analysis by graph coloring, *J. Amer. Statist. Assoc.* 73 (1978) 397–403.
- [6] P. Hansen and B. Jaumard, Minimum sum of diameters clustering, *J. Classification* 4(2) (1987) 215–226.
- [7] L.J. Hubert, Spanning trees and aspects of clustering, *British J. Math. Statist. Psych.* 27 (1974) 14–28.
- [8] B. Leclerc, Caractérisation, construction et dénombrement des ultramétriques supérieures minimales, *Statistiques et Analyse des Données* 11(2) (1986) 26–50.
- [9] C. Monma and S. Suri, Partitioning points and graphs to minimize the maximum or the sum of diameters, *Proc. 6th Internat. Conf. on the Theory and Applications of Graphs* (Wiley, New York, 1989).
- [10] M.R. Rao, Cluster analysis and mathematical programming, *J. Amer. Statist. Assoc.* 66 (1971) 622–626.